

In the Claims

1. (Previously presented) A method of transferring computer software code between a first and a second node in a communications network, each of said nodes comprising a SIP client, said method comprising the steps of:-
 - (i) storing computer software code in a SIP message;
 - (ii) sending the SIP message and computer software code from the first SIP client associated with the first node to the second SIP client associated with the second node; and
 - (iii) executing the computer software code using the second node.
2. (Original) A method as claimed in claim 1 wherein said computer software code is added to the SIP message.
3. (Cancelled)
4. (Cancelled)
5. (Original) A method as claimed in claim 1 wherein said computer software code comprises Java byte code.
6. (Original) A method as claimed in claim 1 wherein said computer software code comprises one or more Java applets.
7. (Original) A method as claimed in claim 1 wherein said computer software code comprises one or more mobile automated software agents.
8. (Previously presented) A method as claimed in claim 7 wherein said mobile automated software agents are Java mobile agents.
9. (Original) A method as claimed in claim 1 wherein said second node comprises a Java virtual machine.

10. (Previously presented) A method as claimed in claim 2 wherein the computer software code is added to the body of the SIP message.
11. (Original) A method as claimed in claim 1 which further comprises adding an indicator to a header of the SIP message in order to indicate the presence of the computer software code and arranging the second SIP client to recognise the indicator.
12. (Original) A method as claimed in claim 1 which further comprises the step of proceeding with any SIP process related to the SIP message.
13. (Previously presented) A method as claimed in claim 11 wherein said second SIP client is arranged such that on receipt of a SIP message containing such an indicator, the computer software code stored in the SIP message is executed by the second node before that second node carries out any other processes related to the SIP message.
14. (Original) A method as claimed in claim 1 wherein said computer software is arranged to interact with the second SIP client via a specified application programming interface.
15. (Previously presented) A method as claimed in claim 1 wherein said computer software is arranged to interact with a processor associated with the second SIP client via a specified application programming interface.
16. (Previously presented) A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to set up a multimedia conference call.

17. (Previously presented) A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to upgrade said SIP client.
18. (Previously presented) A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to carry out a self-test function and to pass the results to another node.
19. (Previously presented) A method as claimed in claim 1 wherein a call is made to the first node, the first node is unable to accept the call and the first node sends the said SIP message and computer software code responsive to being unable to accept the call, said execution of said computer software code causes said second node to accept a forwarded call from the first node and to pass the identity of the second node back to a call originator.
20. (Previously presented) A communications network node comprising:
 - (i) a SIP client;
 - (ii) an input arranged to receive SIP messages;
 - (iii) a processor arranged to extract and execute computer software code from a received SIP message.
21. (Previously presented) A communications network node as claimed in claim 20 wherein said processor comprises a Java virtual machine.
22. (Previously presented) A communications network node as claimed in claim 20 which further comprises an application programming interface arranged to allow the computer software code to interact with the SIP client.
23. (Previously presented) A communications network node as claimed in claim 20 wherein said processor further comprises a detector arranged to detect an indicator in a received SIP message which indicates that computer software code is associated with that SIP message.

24. (Currently amended) A computer program stored in a computer readable medium, the computer program arranged to control a communications network node, said node comprising a SIP client and a processor, said computer program being arranged to control the node when executed on the processor such that when a SIP message is received by the SIP client, which contains computer software code, the software code is executed by the processor.
25. (Original) A computer program as claimed in claim 24 which is stored on a computer readable medium.
26. (Previously presented) A communications network comprising a plurality of communications network nodes each such node comprising:
- (i) a SIP client;
 - (ii) an input arranged to receive SIP messages containing computer software code; and
 - (iii) a processor arranged such that in use, when a SIP message is received, any computer software code contained in that SIP message is executed by the processor.
27. (Previously presented) A method of setting up a conference call between two or more parties, each party comprising a SIP client and a host processor, said method comprising the steps of:
- a) storing computer software code in a SIP message;
 - b) sending the SIP message to each of the parties;
 - c) executing the computer software code at each of the host processors.
28. (Original) A method as claimed in claim 27 wherein the computer software code is arranged to take into account capabilities of each host processor.

29. (Original) A method as claimed in claim 27 wherein said conference call is a multimedia conference call.
30. (Previously presented) A system for automatically setting up a conference call between two or more parties, each party comprising a SIP client and a host processor, said system comprising:- a processor for storing computer software code in a SIP message and to send that SIP message to each of the parties; and wherein each of said host processors is arranged to execute the computer software code in use, when the SIP message is received.
31. (Previously presented) A method of upgrading or replacing interconnected SIP clients each SIP client being associated with a host processor said method comprising the steps of:-
- i) storing computer software code suitable for said upgrade or replacement in a SIP message;
 - ii) sending the SIP message to each of the SIP clients; and
 - iii) executing the computer software at each of the host processors.
32. (Previously presented) A method of testing members of a group of SIP clients each SIP client being associated with a host processor said method comprising the steps of:-
- (i) storing computer software code suitable for said testing in a SIP message;
 - (ii) sending the SIP message one of the SIP clients;
 - (i) executing the computer software at the host processor associated with that SIP client in order to obtain test results; and
 - (ii) repeating steps (ii) to (iii) for each of the other SIP clients in the group.
33. (Previously presented) A method of forwarding a call from a first SIP client to a second SIP client, each of said SIP clients being associated with a host processor, said method comprising the steps of:-

- (ii) receiving a call at the first SIP client and if that call is not answered then storing computer software code in a SIP message, said computer software code being arranged to forward a call;
 - (iii) sending the SIP message from the first SIP client to a specified second SIP client; and
 - (iv) executing the computer software using the host processor associated with the second SIP client such that the call is forwarded to the second SIP client.
34. (Previously presented) A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to replace said SIP client.